

REMARKS/ARGUMENTS

Claims 1 - 54 are pending. The claims have not been amended.

Claim 11 was rejected under 35 U.S.C. § 102(e) for allegedly being anticipated by Riddle et al., U.S. Patent No. 6,457,051.

The following rejections were raised under 35 U.S.C. § 103 for alleged obviousness in view of the associated references:

- claims 1, 2, 4, and 12, Riddle and Tang, U.S. Patent No. 5,378,126;
- claims 3 and 18 - 22, Riddle, Tang, and Moreno, U.S. Patent No. 5,951,674;
- claims 5 - 7, Riddle, Tang, and Del Monte, U.S. Patent No. 5,704,060;
- claims 8, 16, and 17, Riddle, Tang, Del Monte, and Eager et al., U.S. Patent No. 5,960,200
- claim 10, Riddle, Tang, Del Monte, and Moreno;
- claims 13 - 15, Riddle and Del Monte; and
- claims 30 and 37 - 41, Riddle, Tang, Del Monte, and Boucher, U.S. Patent No. 6,226,680.

Claims 23 - 29, 31 - 36, and 42 - 54 were alleged not to add any new limitations to above claims 1 - 22 and claims 37 - 41, and therefore were rejected for similar reasons.

Section 102 Rejection of Independent Claim 11

The present invention is directed to a network switching device that includes classifying network data packets. An aspect of the invention recited in independent claim 11, for example, is "providing one or more regular expressions, each having an associated class identifier" and classifying received data packets by "determining a matching one of said regular expressions that matches said data packet." See also independent claim 47. Riddle et al. was cited for allegedly showing in Table 3 and Fig. 4A the use of regular expressions, and that Table 3 and claim 1 of Riddle et al. allegedly show the step of determining a regular expression that matches a data packet.

Table 3 shows a command language interface to control the functioning of their classifier 304. *Col. 15, lines 52 - 53*. One of ordinary skill in the art will quickly recognize that the commands shown in the command language interface of Table 3 are commands to control

how the classifier functions. The commands are presented in a format that is commonly used in command line interfaces of a computer (e.g., UNIX shell interface, Microsoft DOS interface). The format identifies the parameters of the command.

For example, the command setup **autoclassify {on|off}** is used to cause the classifier to activate (or deactivate) the autoclassification function. Thus, the command “setup autoclassify *on*” activates the function, while “setup autoclassify *off*” would deactivate the function. The commands shown in Table 3 are not regular expressions. The commands shown in Table 3 are commands which control what the classifier will do (e.g., perform autoclassify function), not how to do it. The commands are not used in “determining a matching one of said regular expressions that matches said data packet.”

As for Fig. 4A, the figures shows a flowchart of processing steps for automatically classifying traffic. The flowchart includes a comparison of a flow specification with a “classification tree” (step 404). As Riddle et al. describe,

“A classification tree is a data structure representing the hierarchical aspect of traffic class relationships. Each node of the classification tree represents a class, and has a traffic specification, i.e., a set of attributes or characteristics describing the traffic associated with it. Leaf nodes of the classification tree may contain policies. According to a particular embodiment, the classification process checks at each level if the flow being classified matches the attributes of a given traffic class. If it does, processing continues down to the links associated with that node in the tree.”
Col. 10, lines 27 - 36.

One of ordinary skill would understand that a classification tree is a fundamentally different analytical technique than the technique of regular expressions. It is understood, to one of ordinary skill in the art, that Fig. 4A does not show the use of regular expressions, or that regular expressions are used in “determining a matching one of said regular expressions that matches said data packet.”

Claim 1 of Riddle et al. appears to recite an aspect of their invention as shown in the embodiment of Fig. 4A. Again, the classification tree mentioned in Fig. 4A is fundamentally different analytical tool from Applicant’s regular expression as recited in claim 11. Claim 1 of

Riddle et al. does not teach “determining a matching one of said regular expressions that matches said data packet.”

For at least any one of the foregoing reasons, the Section 102 rejection of claim 11 is believed to be overcome.

Section 103 Rejection of Claims 1, 2, 4, and 12

As mentioned above, the present invention is directed to a switching device that includes classifying network data packets. Another aspect of the invention, as recited in independent claim 1 for example, is “scanning incoming network data using lexical token scanning” to classify a data packet. See also dependent claim 12. The Examiner correctly observes that Riddle et al. do not teach lexical scanning. Tang was cited for teaching a lexical scanning technique. The Examiner asserted it would be obvious to:

“include lexical scanning of packets and parsing lexical tokens in Riddle because doing so would allow the processor to read the specific lexical token relating to the control protocol layer that the instruction belongs to rather than reading the whole instruction or packet and thus create faster communications medium and faster data routing.” *O.A. page 3, last five lines (underlining added).*

A particularly salient aspect of the present invention is the treatment of network data as lexical tokens for the purpose of packet classification. It is respectfully and earnestly submitted that the Examiner improperly applied hindsight knowledge gleaned from the present invention when he characterized the data in Riddle et al. as “specific lexical token” to justify inclusion of the lexical scanning technique shown by Tang.

Neither Riddle et al. nor Tang provide any suggestion, express or otherwise, to subject network data to lexical analysis. Riddle et al. performs classification using a classification tree. One of ordinary skill in the relevant art will understand that classification trees do not represent or suggest the idea of lexical scanning.

Tang is directed to a method for compiling programs. Lexical scanning is a common technique used in compilers for computer programs. Tang describes conventional program compiler techniques for compiling computer program languages. There is no suggestion by Tang to apply the technique of lexical scanning to network data.

Absent teachings from the present invention, there is no suggestion to view network data as lexical tokens, or that network data can be subjected to lexical scanning techniques to perform network data packet classification.

In addition, while the notion to “create faster communications medium and faster data routing” is desirable, it does not suggest how that might be achieved, only that it might be a desirable goal. Thus, creating faster communications and faster data routing does not suggest the specific scanning technique of lexical scanning.

As to claim 2, an aspect of the invention recited therein includes “identifying an arithmetic operation and performing said arithmetic operation” during scanning. Respectfully, it is not at all clear how the cited portion of Riddle et al., repeated below, shows this aspect of the invention:

“In an optional step 410, duplicate instances having the same identifying characteristics are suppressed, in favor of keeping a count of the duplicates and a most recent time traffic with these identifying characteristics was encountered. In an optional step 412, a byte count of traffic of this type has been detected is included. Otherwise, the automatic classification has failed to determine a class and processing returns.” *Col. 14, lines 1 - 9 (underlining added).*

Riddle et al. appear to keep a byte count when a duplicate is detected. Riddle et al. do not describe “identifying an arithmetic operation” during scanning or “performing said arithmetic operation” as a result of identifying the arithmetic operation. The Section 102 rejection of claim 2 is believed to be overcome for at least this additional reason.

As to claim 4, an aspect of the invention recited therein the lexical scanning includes “providing a set of regular expressions.” Excluding the teachings of the present invention, neither Riddle et al. nor Tang suggest that network data can be processed using lexical scanning. It follows therefore that neither reference suggests the use of regular expressions for performing lexical scanning of network data.

Section 103 Rejection of Claims 3 and 18 - 22

An aspect of the present invention recited in claims 3 and 18 - 20 include “identifying a skip operation and in response thereto skipping over one or more subsequent input bytes.” *Claim 3*. The Examiner cited Moreno’s claims 7 and 8 for allegedly showing this aspect of the invention. A distinction between Moreno and the present invention is that Moreno teaches a skip instruction to jump to a different location in the program instruction sequence, not in any input data that the program might be processing. By contrast, the skip operation as recited in claim 3, for example, is for “skipping over one or more subsequent input bytes.”

In addition, the Examiner asserts that the proposed combination of Riddle et al. and Moreno “would allow the processor to detect an error in the stream of incoming data and skip to the next data stream by detecting an address of the beginning of the next incoming data packet.” *O.A., page 5, lines 1 - 5 (underlining added)*. While the branch address in a program instruction sequence is known *a priori* by virtue of compiling source code, such addressing information does not exist in network data packets such as those processed by Riddle et al. as to allow for detecting “the beginning of the next incoming data packet.” A person of ordinary skill will understand that network data packets do not operate in this manner. There is no relation between one packet and another packet such that the beginning of the packet can be determined by performing a skip operation using predetermined branch addresses as taught by Moreno.

As to claim 21, an aspect of the invention recited therein includes “detecting an operator indicating a value to be saved in a register.” Moreno, being a reference directed to program instructions in a VLIW processor, describes register operations. However, Moreno does not suggest modifying Riddle et al. to incorporate a register operation in their flow classification technique.

As to claim 22, an aspect of the invention recited therein includes “detecting an operator indicating a logical or mathematical operation to be performed on the contents of said register.” Moreno, being a reference directed to program instructions in a VLIW processor, describes logical or mathematical operations that are performed on a register. However, Moreno does not suggest modifying Riddle et al. to incorporate into their flow classification technique logical or mathematical operations to be performed on a register.

The Section 103 rejection of claims 3 and 18 - 22 is believed to be overcome for at least the foregoing reasons.

Section 103 Rejection of Claims 5 - 7

An aspect of the invention recited in claims 5 - 7 includes "providing a deterministic finite automaton (DFA) comprising plural states, said step of scanning including recognizing data packets using said DFA." *Claim 5*. Independent claims 23, 31, 40, and 51 also recite this aspect of the invention; for example, "receiving a description of classification rules in a classification language [and] compiling said classification language to produce a deterministic finite automaton (DFA) comprising plural states." *Claim 23*.

As noted above, a particularly relevant aspect of the present invention is the use of lexical scanning techniques to classify network data. Del Monte teaches lexical scanning of text in a document. "The system and method of this invention comprise a lexical parser which divides the text of a document into search terms." *Col. 2, lines 57 - 59*. "[T]he scanner 110 is reset to the beginning of the document 10, so that the lexical parser 120 may use it to scan the document 10." *Col. 13, lines 53 - 55*.

Respectfully, the Examiner has improperly relied on the teaching of the present invention in his assertion that "[o]ne would be motivated to include a DFA including recognizing the lexical tokens using the DFA in Riddle because doing so would allow the DAFT to determine the classification of the data..." *O.A., page 6, second full paragraph (underlining added)*. Characterizing the network data that is processed by Riddle et al. as lexical tokens is precisely that which is taught by the present invention. It is earnestly submitted that improper hindsight reconstruction was employed as the basis for the Section 103 rejection of claims 5 - 7. The rejection is therefore believed to be improper and thus is overcome.

Section 103 Rejection of Claims 8, 16, and 17

An aspect of the present invention recited in claims 8, 16, and 17 is "producing a non-deterministic finite automaton (NFA) from said grammar tree data structure, and converting said NFA to produce said DFA." *Claim 8*. Eager et al. were cited for this aspect of the invention. Dependent claim 8 has independent claim 1 as the base claim. As discussed above for claim 1, a

particularly relevant aspect of the invention is the idea that network data can be viewed as lexical tokens. In order to apply the lexical scanning technique of Tang, the network data of Riddle et al. was characterized as "lexical tokens." It is earnestly submitted that the Examiner was motivated by the teachings of the present invention when he characterized the network data of Riddle et al. as lexical tokens. The proposed combination of Riddle et al. and Tang, in an attempt to obtain an embodiment of the invention recited in claim 1, is therefore based on the use of impermissible hindsight teachings of the present invention.

The Eager et al. reference relates to a system that "transitions an entire enterprise to a distributed infrastructure." *Abstract*. Eager et al. describe NFA's and DFA's in column 5, lines 51 and following. However, Eager et al. do not show, or suggest to one of ordinary skill in the art, the idea taught in the present invention of treating network data as lexical tokens and subjecting the network data to lexical scanning, as recited in claim 1. Consequently, claim 8 is believed to be allowable over the cited art based on the allowability of its independent claim 1.

As to dependent claims 16 and 17, these claims have independent claim 11 as the base claim. As discussed above, Riddle et al. do not show "providing one or more regular expressions [and] determining a matching one of said regular expressions that matches said data packet." *Claim 11*. Table 3 shows the commands of a command language interface to control how the classifier 304 functions. Contrary to the Examiner's assertion, the commands shown in Table 3 are not regular expressions used in "determining a matching one of said regular expressions that matches said data packet." The commands are simply used to control the functioning of the classifier, not how it functions. Neither Tang nor Eager et al. suggest incorporating this aspect of the invention to Riddle et al. Consequently, claims 16 and 17 are believed to be patentable over the cited art based on the allowability of claim 11.

Section 103 Rejection of Claim 10

Claim 10 recites "wherein some of said states [in the DFA] further include a skip instruction." The base claim of claim 10 is claim 1, which as discussed above is not rendered obvious by Riddle et al. and Tang absent the impermissible use of teachings from the present invention, namely, that network data can be viewed as lexical tokens and subject to lexical

scanning, as recited in claim 1. For at least this reason, claim 10 is believed to be allowable, based on the allowability of its base claim.

As discussed above, Moreno teaches a skip instruction to jump to a known location in the program instruction sequence. As one of ordinary skill would understand, and contrary to the Examiner's assertion, network data such as the traffic packets handled by Riddle et al. do not have *a priori* addresses. Therefore Moreno can not be used as a teaching to modify Riddle et al. to arrive at the recited skip instruction of claim 10.

Section 103 Rejection of Claims 13 - 15

Claims 13 - 15 have claim 11 as their base claim. As discussed above, Riddle et al. do not show "providing one or more regular expressions [and] determining a matching one of said regular expressions that matches said data packet." *Claim 11*. Table 3 shows the commands of a command language interface to control how the classifier 304 functions. Contrary to the Examiner's assertion, one of ordinary skill will understand that the commands shown in Table 3 are not regular expressions used in "determining a matching one of said regular expressions that matches said data packet." The commands are simply used to control the functioning of the classifier, not how it functions. Claims 13 - 15 are therefore believed to be allowable over the cited art based on the allowability of their base claim 11.

Section 103 Rejection of Claims 30 and 37 - 41

The independent claim 40 is directed to a network packet classifier. Claim 40 recites a "memory configured with data representing a deterministic finite automaton (DFA) [and] circuits operatively coupled to [the] memory and to [an] input, and configured to lexically scan said data stream with said DFA to produce a reached terminating state." See also independent claims 23, 31, 47, and 51. Tang and Del Monte were cited, respectively, for showing lexical scanning and for showing DFA's. However, the basis for applying Tang and Del Monte to Riddle et al. require that the network data that is processed by Riddle et al. be viewed as lexical packets. As stated above, doing this requires the impermissible hindsight teaching of the present invention. Consequently, claim 40 cannot be obtained in the manner

proposed because there is no teaching outside of the present invention to suggest the proposed combination of prior art references.

Rejection of Claims 23 - 29, 31 - 36 and 42 - 54

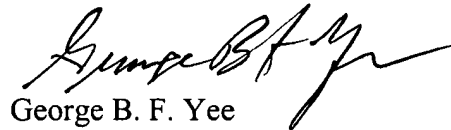
These claims were rejected for allegedly not adding new limitations to the claims 1 - 22 and 37 - 41. For the various reasons set forth above, it is earnestly believed that claims 1 - 22 and 37 - 41 are allowable over the cited art. Therefore, claims 23 - 29, 31 - 36 and 42 - 54 are believed to be allowable over the cited art even if *arguendo* they do not add new limitations to the claims 1 - 22 and 37 - 41, for at least the same reasons that claims 1 - 22 and 37 - 41 are allowable.

CONCLUSION

In view of the foregoing, Applicants believe all claims now pending in this Application are in condition for allowance. The issuance of a formal Notice of Allowance at an early date is respectfully requested.

If the Examiner believes a telephone conference would expedite prosecution of this application, please telephone the undersigned at 650-326-2400.

Respectfully submitted,



George B. F. Yee
Reg. No. 37,478

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, 8th Floor
San Francisco, California 94111-3834
Tel: 650-326-2400 / Fax: 415-576-0300
GBFY:cmm / 60038837 v1